TLe

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

| APPLICATION NO. | FILING DATE | FIRST NAMED INVENTOR | ATTORNEY DOCKET NO. | CONFIRMATION NO. |
|---|---|---|---|---|
| 10/080,793 | 02/20/2002 | Clifford N. Click JR. | SUN1P231C1/P3911 | 2841 |

22434 7590 02/08/2006

BEYER WEAVER & THOMAS LLP
P.O. BOX 70250
OAKLAND, CA 94612-0250

| EXAMINER | |
|---|---|
| PHAM, CHRYSTINE | |
| ART UNIT | PAPER NUMBER |
| 2192 | |

DATE MAILED: 02/08/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

|  |  |  |  |
|---|---|---|---|
| ***Office Action Summary*** | **Application No.**<br>10/080,793 | **Applicant(s)**<br>CLICK ET AL. |  |
|  | **Examiner**<br>Chrystine Pham | **Art Unit**<br>2192 |  |

***-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --***

**Period for Reply**

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

**Status**

1)☒ Responsive to communication(s) filed on *16 November 2005*.

2a)□ This action is **FINAL**. 2b)☒ This action is non-final.

3)□ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

**Disposition of Claims**

4)☒ Claim(s) *1,9,10,12 and 17-23* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)□ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1, 9, 10, 12, 17-23* is/are rejected.

7)□ Claim(s) _____ is/are objected to.

8)□ Claim(s) _____ are subject to restriction and/or election requirement.

**Application Papers**

9)□ The specification is objected to by the Examiner.

10)□ The drawing(s) filed on _____ is/are: a)□ accepted or b)□ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)□ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

**Priority under 35 U.S.C. § 119**

12)□ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)□ All b)□ Some * c)□ None of:

1.□ Certified copies of the priority documents have been received.

2.□ Certified copies of the priority documents have been received in Application No. _____.

3.□ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

**Attachment(s)**

1) □ Notice of References Cited (PTO-892)

2) □ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) □ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08)
Paper No(s)/Mail Date _____.

4) □ Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____.

5) □ Notice of Informal Patent Application (PTO-152)

6) □ Other: _____.

## DETAILED ACTION

1. A request for continued examination under 37 CFR 1.114, including the fee set forth in 37 CFR 1.17(e), was filed in this application after final rejection. Since this application is eligible for continued examination under 37 CFR 1.114, and the fee set forth in 37 CFR 1.17(e) has been timely paid, the finality of the previous Office action has been withdrawn pursuant to 37 CFR 1.114. Applicant's submission filed on November 16th 2005 has been entered.

2. This action is responsive to Amendment filed on November 16th 2005. Claim 1 has been amended. Claims 2-8, 11, 13-16 have been canceled. Claims 17-23 are new claims. Claims 1, 9, 10, 12, 17-23 are pending.

### *Claim Objections*

3. Claims 1 and 17 are objected to because they contain the following informalities: "complied code" recited in claim 17 (line 2) and claim 1 (4th to last line) should be replaced with --compiled code--.

### *Response to Arguments*

4. Applicants' arguments filed November 16th 2005 have been fully considered but they are not persuasive.

First, Applicants contend, "McQuistan et al. does NOT teach or suggest an adapter that can facilitate translation of an execution stack" (Remarks, page 9 of 10, 2nd paragraph). Applicants essentially contend that the argument stack created at runtime by the interpreter of McQuistan et al. (hereinafter, "McQuistan") does not teach, "translating an execution stack that can be used for compiled code" (Remarks, page 9 of 10, 2nd paragraph). The Examiner respectfully disagrees.

- As has been established in the final Office Action (page 8), in FIG.4 and col.6:47-67, McQuistan specifically disclose the client stubs 408 and 412 interfacing with the interpreter 404. The interpreter 404 **marshals** (i.e., translate/convert) **arguments** (of the stubs) into a **runtime buffer** maintained by the RPC runtime 402. The interpreter 404 also **unmarshals** (i.e., translate/convert) **arguments** out of the **runtime buffer** and passes the arguments to the client **stubs**. In FIG.5 and col.7:40-col.8:50, McQuistan expressly discloses the process of a client (i.e., caller) invoking a remote procedure residing on the server wherein the arguments of the client (for invoking/calling a remote procedure) are pushed onto an stub argument stack, which is then passed to the server. After receiving the argument stack from the client, the server's interpreter marshals (i.e., translate/convert) the arguments into the runtime buffer, which is then passed to the RPC runtime facility on the server to carry out the requested [remote] procedure. The output of the [remote] procedure is then pushed onto the runtime buffer and passed back to the server's interpreter. The server's interpreter than unmarshals (i.e., translate/convert) the output stored from the runtime buffer into the stub argument stack of the client. In col.2:1-20, McQuistan expressly discloses a **buffer as a stack** that contains arguments in a specific order. As admitted by Applicants, col.7:4-25 of McQuistan also teaches creating an argument stack at runtime (i.e., during execution). In the same passage, McQuistan expressly discloses the client side interpreter and the server side interpreter are stored in dynamically linked library and are linked into the address space of the caller (e.g., server) or callee (e.g., client) at runtime. Needless to say, this clearly anticipates that the remote procedure on the server is invoked during execution of the caller application on the client. Thus, it is clear from these passages that the both the argument stack and the runtime buffer are execution stacks utilized by the client runtime environment and the server runtime environment, respectively. Moreover, the marshaling of the argument stack created at runtime (for the caller/client) into the runtime buffer/stack anticipates translating the execution stack. McQuistan does not expressly disclose the interpreter as being associated with the virtual machine. However, *Pelegri* already teaches a virtual machine, which is well known in the art to be a stack-based machine. Thus, it is obvious that the interpreters of McQuistan are virtual machines since they rely on runtime stacks to perform the marshaling and unmarshaling functions.

Second, Applicants contend, "The cited art does NOT teach or suggest an adapter/stub that can behave as an adapter or stub for a virtual machine" (Remarks, page 9 of 10, last 2 paragraphs). The Examiner respectfully disagrees.

- ➢ As recited in claim 1, "providing an interpreter to compiled code (I/C) **adapter** that facilitates translation of a first execution stack used by an interpreter … so that the first execution stack can be subsequently be used to execute compiled code compiled by a compiler associated with the virtual machine" (Emphasis added). As discussed above, McQuistan teaches unmarshaling (i.e., translating) the runtime buffer/stack (containing the result of the invoked remote procedure) (i.e., "first execution stack") into the argument stack used by the client application (i.e., caller of the remote procedure). It is respectfully submitted that McQuistan's client stub (for invoking the remote procedure) clearly anticipates an I/C adapter that **facilitates** the translation of the execution stack (i.e., runtime buffer/stack) used by the interpreter, that is to say, the client stub is also an adapter, because without the client stub/adapter, there can be no argument stack, from which a runtime buffer can be allocated. Without the client stub/adapter, the remote procedure (i.e., callee) cannot be invoked by the client application (i.e., caller). Thus, contrary to Applicants' argument, the client stub clearly anticipates the adapter that facilitates the translation of the execution stack.

Lastly, Applicants contend, "the cited art does NOT teach or suggest determining whether to provide an interpreter to compiled code (I/C adapter) or a compiled code to interpreter (C/I) adapter" (Remarks, page 10 of 10, 1st paragraph). The Examiner respectfully disagrees.

➢ As recited in claim 1, "providing a compiled code to interpreter (C/I) adapter that **facilitates** translation of a second execution stack used for execution of compiled code ... so that the second execution stack can be subsequently be used by an interpreter associated with the virtual machine". As discussed above, McQuistan teaches **marshaling** (i.e., translating) the **stub argument stack** [used by the executing client application] (i.e., "second execution stack") into the **runtime buffer/stack** used by the interpreter and the remote procedure. Furthermore, col.6:25-28 of McQuistan expressly discloses compiling the client stub code to produce compiled client stub to be executed along with the client application. Thus, it is respectfully submitted that, McQuistan's client stub clearly anticipates an C/I adapter that **facilitates** the translation of stub argument stack (i.e., "second execution stack") used for execution of compiled code compiled by a compiler, so that the second execution stack can be subsequently used (as runtime buffer/stack) by an interpreter. Since McQuistan interpreter marshals an argument stack (i.e., providing compiled code to interpreter C/I adapter) only **in response** to being invoked by a client stub, and unmarshals the runtime buffer (i.e., providing interpreter to compiled code I/C adapter) only **in response** to being invoked by the remote procedure (upon being completed), McQuistan clearly teaches determining whether to provide an I/C adapter or C/I adapter.

### *Claim Rejections - 35 USC § 101*

5. 35 U.S.C. 101 reads as follows:

> *Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.*

Claims 18-20 are rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter. Claims 18-20 recites "a computer readable medium including computer program code". On page 9, 2nd paragraph, the specification describes the computer readable medium as "data signal embodied in a carrier wave", which does not limit the claimed "product" to **tangible** products and media. Moreover, it does not appear a claim reciting a signal encoded with functional descriptive material falls within **any** of the categories (i.e., manufacture) of patentable subject matter set forth in § 101. See the Interim Guidelines for Examination of Patent Applications for Subject Matter Eligibility, signed on October 26, 2005 – OG Cite: 1300 OG 142 (<http://www.uspto.gov/web/offices/com/sol/og/2005/week47/patgupa.htm>).

### *Claim Rejections - 35 USC § 103*

6. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> *(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.*

7. Claims 1, 9, 10, 12, 17-23 are rejected under 35 U.S.C. 103(a) as being unpatentable over *Pelegri* in view of McQuistan et al. (*McQuistan*, US 6321275 B1).

**Claim 1**

*Pelegri* teaches in a computer system (see at least FIG.3 & associated text), a method for generating an adapter/stub (see at least *run-time, new stub class* col.6:33-65) for a virtual machine (see at least *11, 15* FIG.1 & associated text; *11, 15* Fig.3 & associated text; *virtual machine, local machine, remote machine* col.6:42-50) during runtime (see at least *410* FIG.3 & associated text; *816* Fig.10 & associated text; *910* Fig.11 & associated text), comprising:

- identifying a machine state input parameter for a machine state (see at least *stub class, remote object, second virtual machine* col.4:15-55);
- identifying input parameters for a call to compiled code (see at least *clients, object handles, remote objects, stub objects* col.2:50-67);

- mapping the machine state input parameter and the machine state to the input parameters for the call to compiled code (see at least *stub class, remote object, second virtual machine* col.4:15-55); and
- mapping the machine state and a return value to an exit point of an interpreter to compiled code adapter (see at least *virtual machine 11, 15, stub 60, object 62, remote object 1* FIG.1 & associated text).
- providing a stub representation to a compiler for compilation (see at least *410* FIG.3 & associated text); and
- generating object code base upon the compilation (see at least *run-time stub 60* col.6:33-65; *Java* col.13:50-55).

*Pelegri* further teaches wherein the adapter/stub is a platform-specific interpreter to compiled code (I/C) adapter/stub (see at least *object handles, remote objects, process, remote machine, local machine, stub objects* col.2:50-67). *Pelegri* does not expressly disclose wherein the adapter/stub can behave as an adapter or a stub for the virtual machine; [determining whether to] generate an adapter/stub that can behave as an adapter or stub for the virtual machine; determining whether to provide an interpreter to compiled code (I/C) adapter or a compiled code to interpreter (C/I) adapter when the determining determines to provide the adapter/stub as an adapter; providing an interpreter to compiled code (I/C) adapter that facilitates translation of a first execution stack used by an interpreter associated with the virtual machine when the determining determines to provide the (I/C) adapter, so that the first execution stack can subsequently be used to execute compiled-code compiled by a compiler associated with the virtual machine; and providing a compiled code to interpreter (C/I) adapter that facilitates translation of a second execution stack used for execution of compiled code compiled by a compiler associated with the virtual machine when the determining determines to provide the C/I adapter, so that the second execution stack can be subsequently be used by an interpreter associated with the virtual machine.

However, *McQuistan* discloses

- o wherein the adapter/stub can behave as an adapter or a stub for the virtual machine (see at least FIG.5 & associated text; col.7:40-col.8:50, FIG.4 & associated text; col.6:47-67);
- o [determining whether to] generate an adapter/stub that can behave as an adapter or stub for the virtual machine (see at least FIG.5 & associated text; col.7:40-col.8:50, FIG.4 & associated text; col.6:47-67);
- o determining whether to provide an interpreter to compiled code (I/C) adapter or a compiled code to interpreter (C/I) adapter when the determining determines to provide the adapter/stub as an adapter (see at least FIG.5 & associated text; col.7:40-col.8:50, FIG.4 & associated text; col.6:47-67);
- o providing an interpreter to compiled code (I/C) adapter that facilitates translation of a first execution stack used by an interpreter associated with the virtual machine when the determining determines to provide the (I/C) adapter, so that the first execution stack can subsequently be used to execute compiled-code compiled by a compiler associated with the virtual machine (see at least FIG.5 & associated text; col.7:40-col.8:50, FIG.4 & associated text; *unmarshaling* col.6:47-67); and

- o providing a compiled code to interpreter (C/I) adapter that facilitates translation of a second execution stack used for execution of compiled code compiled by a compiler associated with the virtual machine when the determining determines to provide the C/I adapter, so that the second execution stack can be subsequently be used by an interpreter associated with the virtual machine (see at least FIG.5 & associated text; col.7:40-col.8:50, FIG.4 & associated text; *marshaling* col.6:47-67).

*Pelegri* and *McQuistan* are analogous art because they are both directed to compiling adapter/stub code. It would have been obvious to one of ordinary skill in the pertinent art at the time the invention was made to incorporate the teaching of *McQuistan* into that of *Pelegri* for the inclusion of translating the execution stack. And the motivation for doing so would have been to enable invocation of the remote object (i.e., function) by translating data from a format acceptable to a communication mechanism to a format acceptable to a the function at runtime (see *McQuistan* col.1:65-col.2:22; col.4:13-45).

**Claim 9**

The rejection of base claim 1 is incorporated. *Pelegri* further teaches wherein the method is performed in response to a determination that the adapter/stub is not stored in an adapter/stub library associated with the computer system (see at least *618* FIG.8 & associated text; *stub class cache check unit 614, stub class cache 618, stub class generator 620* col.9:65-col.10:13).

**Claim 10**

The rejection of base claim 9 is incorporated. *Pelegri* further teaches wherein the determination is performed when compiled code is to be executed by the computer system (see at least *client application 9, stub 60, object handle 62, remote object 1* FIG.1 & associated text), and the computer system determines that an interpreter to compiled code (I/C) adapter/stub is required (see at least *618* FIG.8 & associated text; *stub class cache check unit 614, stub class cache 618, stub class generator 620* col.9:65-col.10:13).

**Claim 12**

The rejection of base claim 1 is incorporated. *Pelegri* further teaches wherein the adapter/stub is further operable to update the states of different components of the computer system (see at least *objects, state, class, member functions* col.1:37-col.2:2; *clients, object handles, remote objects, member functions, stub objects* col.2:50-67).

**Claim 17**

The rejection of base claim 1 is incorporated. *Pelegri* as modified by *McQuistan* further teaches wherein said determining of whether to provide an I/C adapter or a C/I adapter comprises: determining whether one or more bytecodes have been processed by an interpreter (see at least *McQuistan* FIG.5 & associated text; col.7:40-col.8:50, FIG.4 & associated text; col.6:47-67).

**Claims 18-20**

Claims recite a computer readable medium including computer program code for performing the method addressed in claims 1, and 17, therefore, are rejected for the same reasons cited in claims 1 and 17.

**Claims 21-23**

Claims recite a computing system comprising at least one processor that performs the method addressed in claims 1, and 17, therefore, are rejected for the same reasons cited in claims 1 and 17.

### *Conclusion*

8. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Chrystine Pham whose telephone number is 571-272-3702. The examiner can normally be reached on Mon-Fri, 8:30am-5pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Tuan Q. Dam can be reached on 571-272-3695. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

CP
February 5, 2006

TUAN DAM
SUPERVISORY PATENT EXAMINER